

武汉东湖学院计算机科学学院

课程 设计 报 告

课程名称_____数据结构_____

题 目_____武汉东湖学院校园导游系统_____

专业班级_____24 级网络空间安全一班_____

学 号_____2024040731192_____

学生姓名_____李畅_____

指导教师_____丰洪才_____

_____2025 年 12 月 12 日_____

武汉东湖学院计算机科学学院

课程设计任务书

课程名称：_____ 数据结构 _____

设计题目：_____ 武汉东湖学院校园导游系统 _____

专 业：_____ 网络空间安全 _____ 班 级：_____ 2024 级 一班 _____

完成时间：_____ 2025.12.12 _____ 指导教师：_____ 丰洪才 _____ 专业负责人：_____

主要内容

设计一个学校校园导游系统，实现学校校园导览系统，完成学校校园景点信息的存储与查询功能，运用遍历算法实现学校校园景点间游览线路的查询与规划功能，运用最短路径算法实现学校校园内任意两景点间的最短路径查询功能，运用最小生成树算法解决学校校园内管道系统的最优布置问题。

基本要求：

- 1、学校的景点地点（景点名称、景点介绍）不少于 8 个，交通道路不少于 10 条，灵活的实现景点地点和道路的扩充及撤销；
- 2、实现图的邻接矩阵与邻接表两种存储结构；
- 3、导游系统之游览线路规划，使用图的两种基本遍历算法：深度优先搜索(DFS)与广度优先搜索(BFS)，运用遍历算法实现景点间游览线路的查询与规划功能；
- 4、导游系统之景点间最短路径规划，使用迪杰斯特拉(Dijkstra)算法和弗洛伊德(Floyd)算法，求最短路径；
- 5、导游系统之管道优化布置规划，使用普里姆(Prim)算法和克鲁斯卡尔(Kruskal)算法，运用最小生成树算法解决管道系统的最优布置问题。

参考资料

[1]李春葆主著，数据结构教程（C/C++版），清华大学出版社

[2]李春葆主著，数据结构上机实验教程（C/C++版），清华大学出版社

武汉东湖学院计算机科学学院

课程设计成绩评价表

课程名称		数据结构						
题 目		武汉东湖学院校园导游系统						
学生姓名		李畅	学号	2024040731192	指导教师姓名	丰洪才	职称	教授
序 号	评价项目	指						

指导教师： 丰洪才

2025 年 月 日

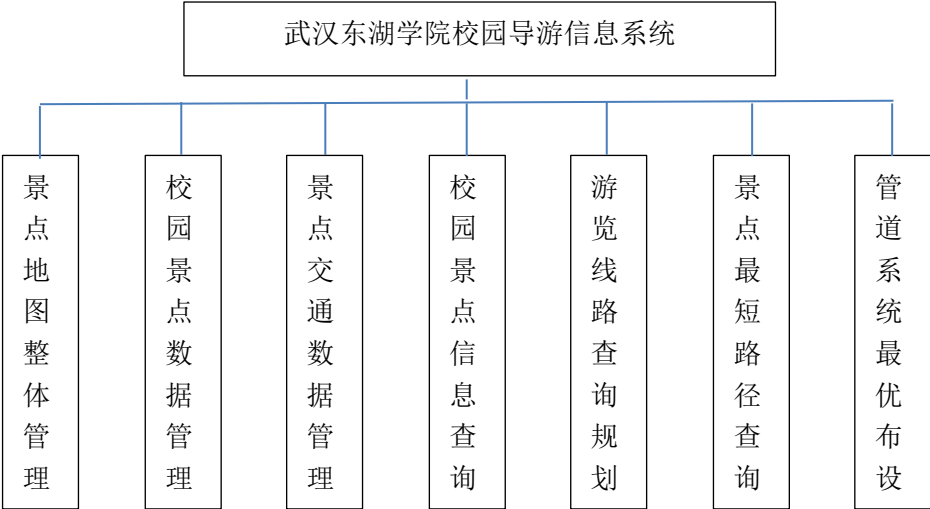
1. 需求分析

编写程序，设计一个学校校园导游系统，实现学校校园导览图系统，完成学校校园景点信息的存储与查询功能，运用遍历算法实现学校校园景点间游览线路的查询与规划功能，运用最短路径算法实现学校校园内任意两景点间的最短路径查询功能，运用最小生成树算法解决学校校园内管道系统的最优布设问题。基本要求包括如下几个方面：

- （1）、学校的景点地点（景点名称、景点介绍）不少于 8 个，交通道路不少于 10 条，灵活的实现景点地点和道路的扩充及撤销；
- （2）、实现图的邻接矩阵与邻接表两种存储结构；
- （3）、导游系统之游览线路规划，使用图的两种基本遍历算法：深度优先搜索(DFS)与广度优先搜索(BFS)，运用遍历算法实现景点间游览线路的查询与规划功能；
- （4）、导游系统之景点间最短路径规划，使用迪杰斯特拉(Dijkstra)算法和弗洛伊德(Floyd)算法，求最短路径；
- （5）、导游系统之管道优化布设规划，使用普里姆(Prim)算法和克鲁斯卡尔(Kruskal)算法，运用最小生成树算法解决管道系统的最优布设问题。

2. 概要设计

武汉东湖学院校园导游信息系统主要由以下七个模块组成，包括：景点地图整体管理、校园景点数据管理、景点交通数据管理、校园景点信息查询、游览线路查询规划、景点最短路径查询、管道系统最优布设。



3. 详细设计

(1) 数据结构体设计

为了实现武汉东湖学院导游系统的管理，采用的主要数据结构为邻接矩阵、邻接表两种存储结构。

① 符常量的定义

```
#define INF 32767          //定义无穷大
#define MAX 10             //最大顶点个数
```

② 以下定义图的邻接矩阵类型

```
typedef struct {
    char SpotId[2];          //景点编号
    char Name[20];           //景点名称
    char Introduction[256];   //景点简介
} VertexType;               //景点类型
typedef struct {
    int u;                   //边的起始顶点
    int v;                   //边的终止顶点
    int Weight;              //边的权重
} EdgeType;                 //边的类型
typedef struct {
    int VexNum,EdgeNum;      //顶点数，边数
    VertexType Vexs[MAXVEXNUM]; //存放顶点信息
    int Edges[MAXVEXNUM][MAXVEXNUM]; //邻接矩阵数组
} MatrixGraph;              //完整的图邻接矩阵类型
```

③ 以下定义邻接表类型

```
typedef struct ANode
{
    int adjvex;              //该边的邻接点编号
    struct ANode *nextarc;    //指向下一条边的指针
    int weight;               //该边的相关信息，如权值（用整型表示）
} ArcNode;                  //边节点类型
typedef struct Vnode
{
    VertexType VexData;      //顶点数据
    int count;               //存放顶点入度,仅仅用于拓扑排序
    ArcNode *firstarc;       //指向第一条边
} VertexNode;               //邻接表头节点类型
typedef struct
{
    VertexNode adjlist[MAXVEXNUM]; //邻接表头节点数组
    int n,e;                   //图中顶点数 n 和边数 e
} AdjGraph;                  //完整的图邻接表类型
```

(2) 系统界面设计

① 主界面设计

武汉东湖学院校园导游信息系统主菜单

- 1 景点地图整体管理
- 2 校园景点数据管理
- 3 景点交通数据管理
- 4 校园景点信息查询
- 5 游览线路查询规划
- 6 景点最短路径查询
- 7 管道系统最优布设
- 0 退出校园导游系统

请选择(0~7):

② 景点地图整体管理界面设计

校园景点地图整体数据管理

1 初始化景点地图数据
2 保存景点地图数据
3 读取景点地图数据
0 返回主菜单
#####

③ 校园景点数据管理界面设计

=====

校园景点数据管理

=====

1 增加景点数据
2 删除景点数据
3 修改景点数据
4 显示景点数据
0 返回主菜单

=====

④ 景点交通数据管理界面设计

~~~~~

校园景点交通数据管理

~~~~~

1 增加景点交通数据
2 删除景点交通数据
3 修改景点交通数据
4 显示景点交通数据
0 返回主菜单

~~~~~

⑤ 校园景点信息查询

⑥ 游览线路查询规划界面设计

#####  
校园景点游览线路查询规划  
#####  
1 深度优先游览线路查询规划  
2 广度优先游览线路查询规划  
0 返回主菜单  
#####

### ⑦ 景点最短路径查询界面设计

```

校园景点景点最短路径查询
1 迪杰斯特拉(Dijkstra)算法
2 弗洛伊德(Floyd) 算法
0 返回主菜单

```

### ⑧管道系统最优布设界面设计

```
#####
                      校园管道系统最优布设
#####
1  普 里 姆( Prim )算法
2  克鲁斯卡尔(Kruskal)算法
0  返回主菜单
#####
```

#### 4、函数及调用关系

主程序 `main()` 函数，在其中定义 `MatrixGraph` 类型变量 `g`，调用 `void ShowMainMenu()` 函数，显示武汉东湖学院导游系统程序主菜单函数，用户选择程序主菜单编号，调用相应的函数实现对应功能；

(1) 用户选择程序主菜单编号 1, 主程序调用景点地图整体管理函数, void ManaMapData(MatrixGraph &g), 通过显示“校园景点地图整体数据管理”菜单界面, 实现景点地图整体管理功能。在这个菜单界面中, ①调用 void InitSpotsData(MatrixGraph &g) 函数, 完成初始化学校景点数据, ②调用 void SaveToMapFile(MatrixGraph g) 函数, 保存景点地图数据到文件“CampusSpotsData.dat”, ③调用 void LoadFromMapFile(MatrixGraph &g) 函数从文件“CampusSpotsData.dat”中读取景点地图数据。

(2) 用户选择程序主菜单编号 2, 主程序调用校园景点数据管理函数, void ManaVertexData(MatrixGraph &g), 通过显示“校园景点数据管理”菜单界面, 实现校园景点数据管理功能。在这个菜单界面中, ①调用 void InMultSpotsData(MatrixGraph &g,int n)函数, 完成输入多个景点信息功能, 其中, 该函数调用 void InOneSpotsData(MatrixGraph &g,int i), 输入一个景点信息, ②调用 void InDeleteSpotsData(MatrixGraph &g) 函数, 删除一个景点信息, ③调用 void InModifySpotsData(MatrixGraph &g) 函数, 修改景点信息, ④调用 void ShowAllSpotsData(MatrixGraph g)函数显示全部景点信息。

(3) 用户选择程序主菜单编号 3, 主程序调用景点交通数据管理函数, void ManaRoadData(MatrixGraph &g), 通过显示“校园景点交通数据管理”菜单界面, 实现校园景点交通数据管理功能。在这个菜单界面中,

①调用 void AddEdge(MatrixGraph &g)函数, 增加一个景点交通数据, 其中该函数调用 int LocateSpotPos(MatrixGraph g, char SpotName[])函数, 完成按景点名称定位存储下标的功能;

②调用 void DeleteEdge(MatrixGraph &g)函数，删除一个景点交通数据，其中该函数调用 int LocateSpotPos(MatrixGraph g , char SpotName[])函数，完成按景点名称定位存储下标的功能；  
③调用 void ModifyEdge(MatrixGraph &g)函数，修改景点交通数据，其中该函数调用 int LocateSpotPos(MatrixGraph g , char SpotName[])函数，完成按景点名称定位存储下标的功能；  
④调用 void DispAllEdge(MatrixGraph g) 函数，显示所有景点交通数据。

（4）用户选择程序主菜单编号 4，主程序调用校园景点信息查询函数，void SearchSpotInfor(MatrixGraph g)，完成游览线路查询，其中该函数调用 int LocateSpotPos(MatrixGraph g , char SpotName[])函数，按景点名称定位存储下标，调用 void ShowOneSpotsData(MatrixGraph g,int i)函数显示一个景点的信息。

（5）用户选择程序主菜单编号 5，主程序调用景点交通数据管理函数，void VisitPathPlan(MatrixGraph &g)，该函数首先调用 bool MatrixToAdjList(MatrixGraph g, AdjGraph \*&G) 函数,通过图的邻接矩阵创建图的邻接表,并调用 void DispAdj(AdjGraph \*G) 函数显示邻接表,以验证创建邻接表是否成功;在提示“请输入游览线路起点景点名称”后,调用 int LocateSpotPos(MatrixGraph g , char SpotName[])函数,按景点名称定位存储下标,判断起点景点正确的前提下,通过显示“游览线路查询规划”菜单界面,实现校园景点游览线路查询规划功能,在这个菜单界面中,

①调用 void DFS(AdjGraph \*G,int v) 函数，按照深度优先遍历算法，完成校园景点游览线路查询规划的功能；

②调用 void BFS(AdjGraph \*G,int v)函数，按照广度优先遍历算法，完成校园景点游览线路查询规划的功能；

③ 调用 void DestroyAdj(AdjGraph \*&G)函数，在销毁图的邻接表后返回主菜单。

（6）用户选择程序主菜单编号 6，主程序调用景点最短路径查询函数，void ShortestPathQuery(MatrixGraph g)，通过显示“景点最短路径查询”菜单界面，实现校园景点游览线路查询规划功能，在这个菜单界面中，

①调用 void Dijkstra(MatrixGraph g, int v) 函数，按照迪杰斯特拉算法，在“请输入游览线路起点景点名称:”后，调用 int LocateSpotPos(MatrixGraph g , char SpotName[])函数，按景点名称定位存储下标，判断起点景点正确的前提下，完成校园景点最短路径查询的功能，其最后该函数调用 void Disppath(MatrixGraph g , int dist[], int path[], int S[] , int v) 函数输出最短路径；

②调用 void Floyd(MatrixGraph g) 函数，按照 Floyd 算法，完成校园景点最短路径查询的功能，其间该函数调用 void Dispath(MatrixGraph g, int A[][MAXVEXNUM], int path[][MAXVEXNUM])函数输出最短路径；



(7) 用户选择程序主菜单编号 7，主程序调用 void PipelineSystemOptimal(MatrixGraph g) 函数，通过显示“管道系统最优布设”菜单界面，实现校园景点管道系统最优布设功能，在这个菜单界面中，

①调用 void Prim(MatrixGraph g,int v) 函数，按照求顶点 v 出发的最小生成树 Prim 算法，在“请输入管道系统最优布设起点景点名称:”后，调用 int LocateSpotPos(MatrixGraph g , char SpotName[])函数，按景点名称定位存储下标，判断起点景点正确的前提下，完成校园景点管道系统最优布设的功能；

②调用 void Kruskal(MatrixGraph g) 函数，按照求最小生成树的 Kruskal 算法，完成校园景点管道系统最优布设的功能，其间该函数调用冒泡排序函数 void BubbleSort(EdgeType R[], int n) ，按边的权重进行了升序排序；

(0) 用户选择程序菜单编号 0，主程序退出管理系统。

4、源代码

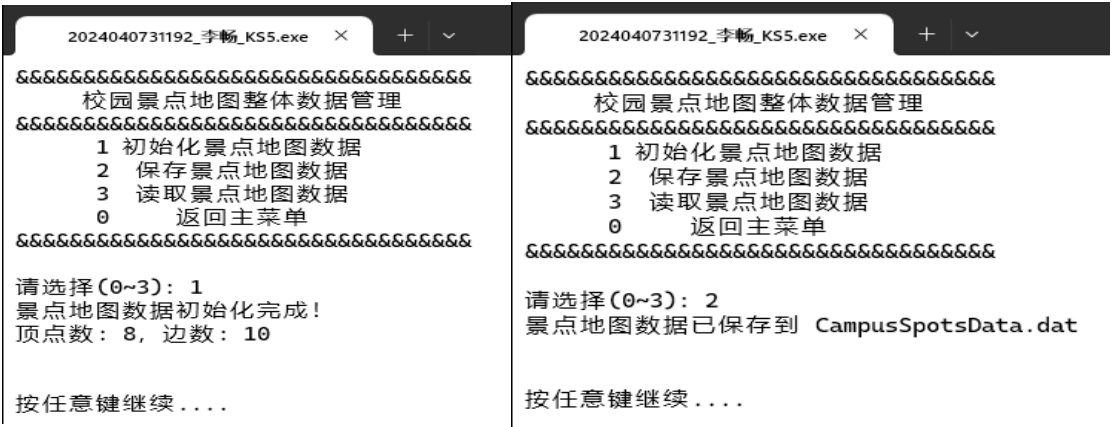
源代码请见电子版！

5、各项功能测试截图

(1) 景点地图整体管理

①初始化景点地图数据功能截图

②保存景点地图数据功能截图

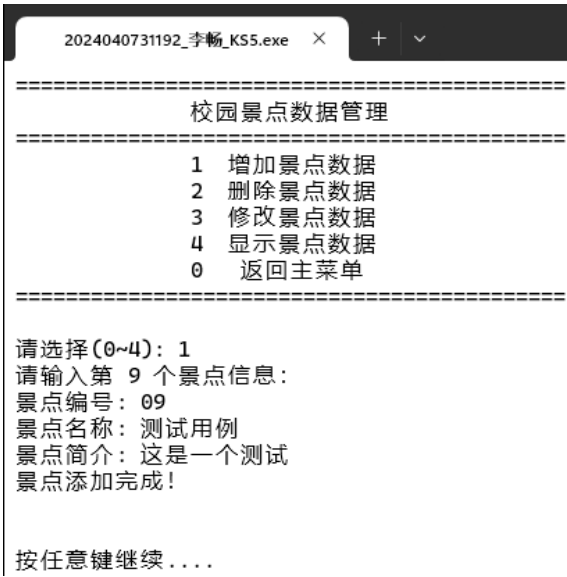


③读取景点地图数据功能截图

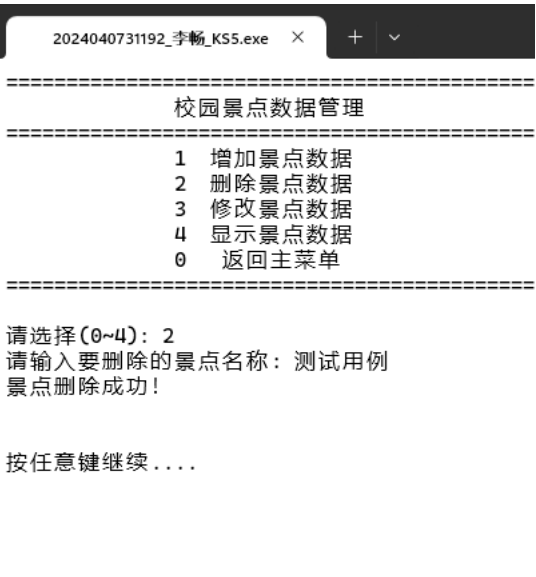


(2) 校园景点数据管理

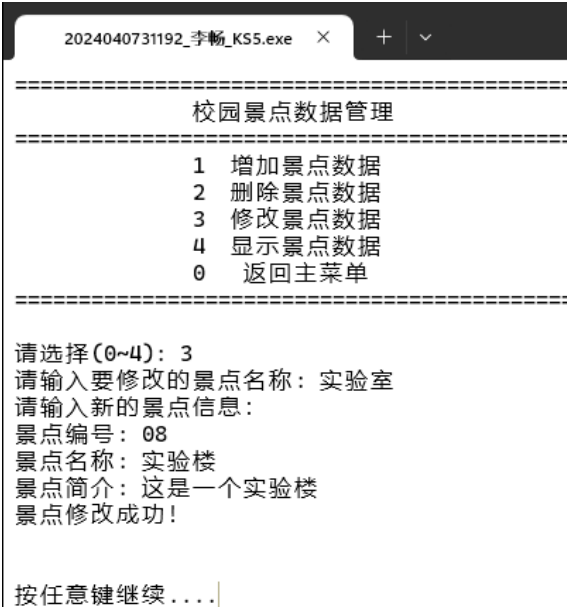
①增加景点数据功能截图



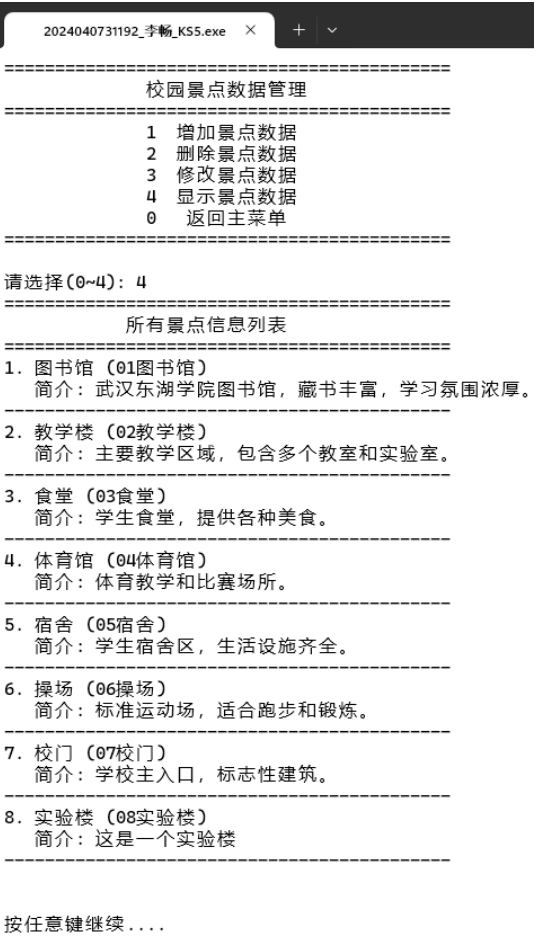
②删除景点数据功能截图



③修改景点数据功能截图

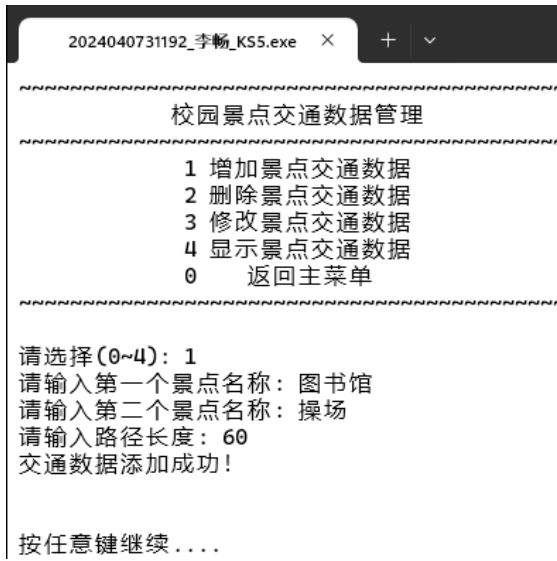


④显示景点数据功能截图

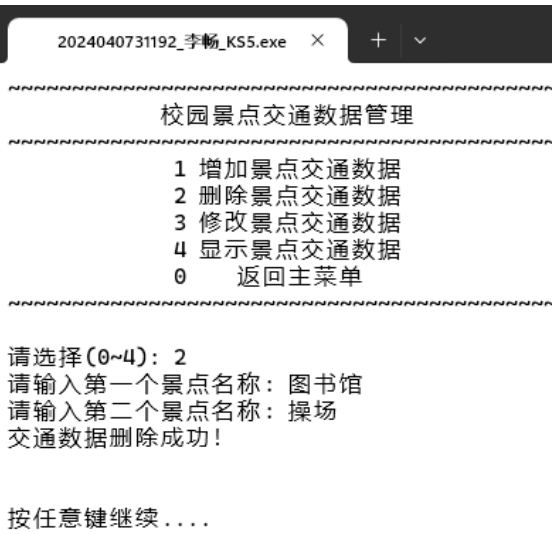


(3) 景点交通数据管理

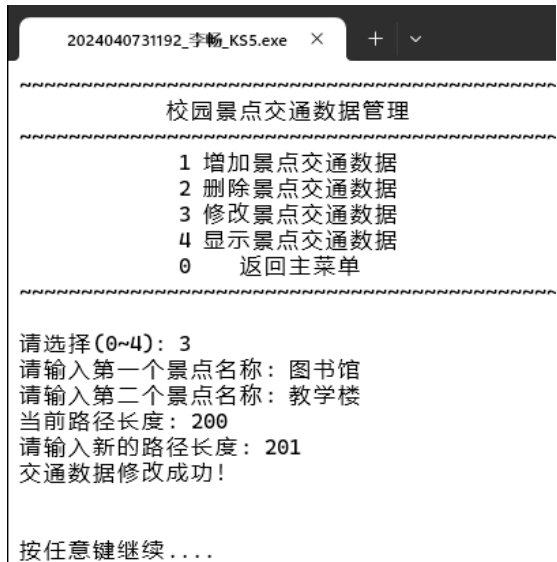
①增加景点交通数据功能截图



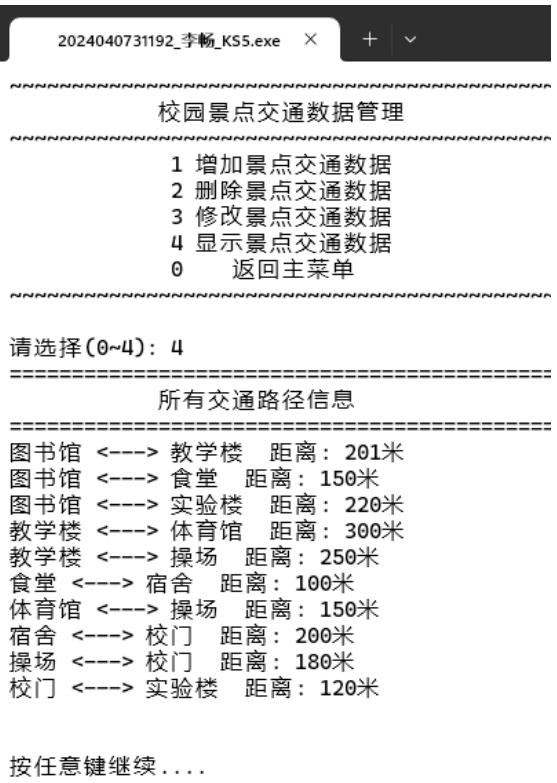
②删除景点交通数据功能截图



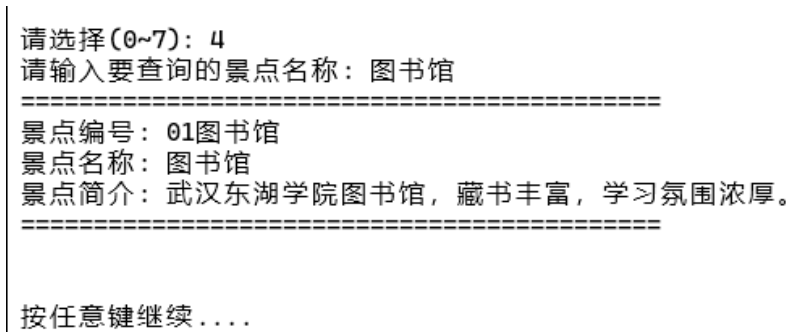
③修改景点交通数据功能截图



④显示景点交通数据功能截图

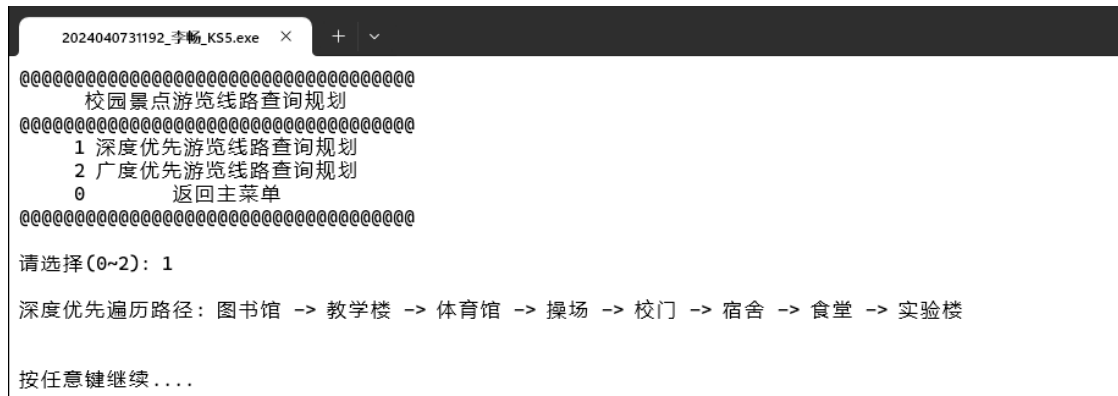


(4) 校园景点信息查询

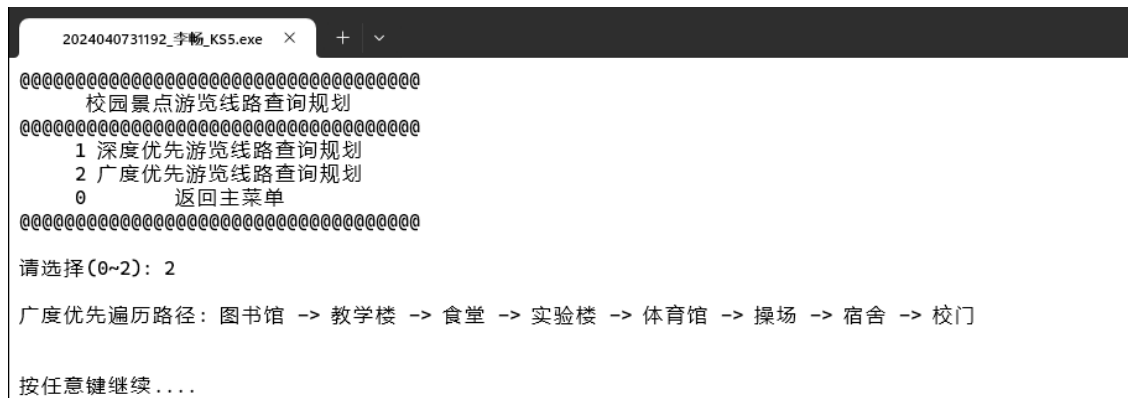


## （5）游览线路查询规划

### ①深度优先游览线路查询规划功能截图

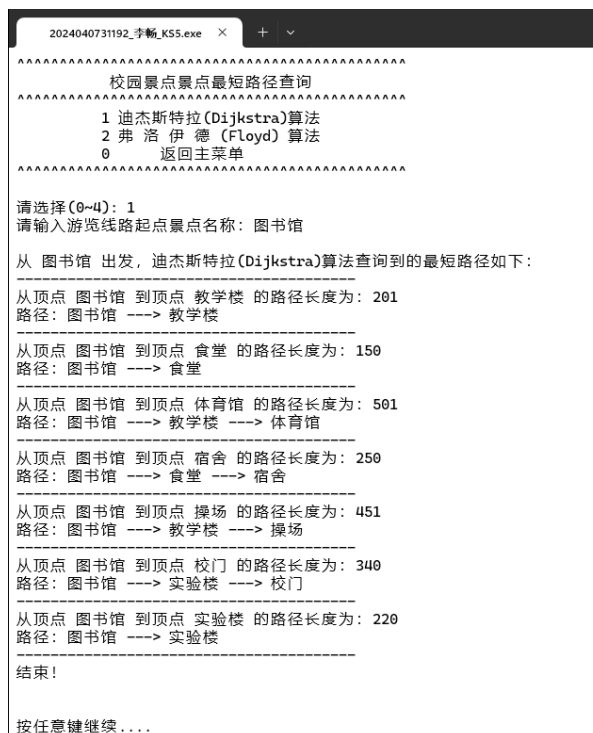


### ②广度优先游览线路查询规划功能截图

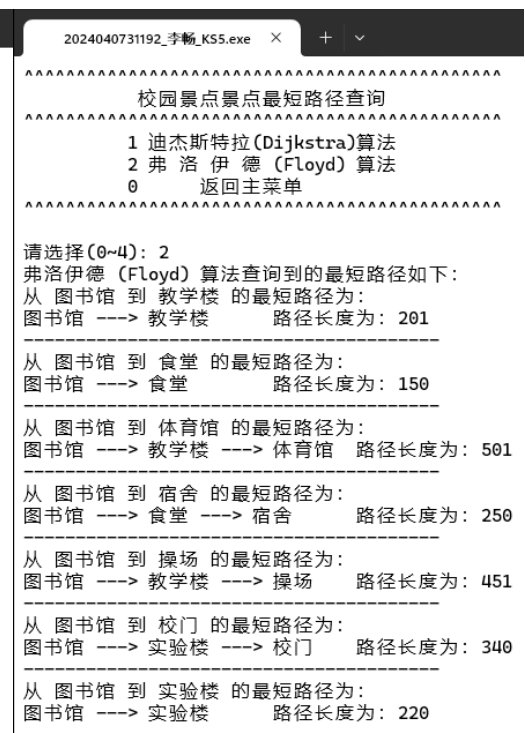


## （6）景点最短路径查询

### ①迪杰斯特拉(Dijkstra)算法功能截图



### ②弗洛伊德 (Floyd) 算法功能截图



## （7）管道系统最优布设

①普里姆(Prim)算法功能截图

```
#####
                校园管道系统最优布设
#####
                1 普里姆(Prim)算法
                2 克鲁斯卡尔(Kruskal)算法
                0  返回主菜单
#####

请选择(0~2): 1
请输入管道系统最优布设起点景点名称: 图书馆

普里姆(Prim)算法最小生成树:
=====
图书馆 --- 食堂  权重: 150
食堂 --- 宿舍  权重: 100
宿舍 --- 校门  权重: 200
校门 --- 实验楼 权重: 120
校门 --- 操场  权重: 180
操场 --- 体育馆 权重: 150
图书馆 --- 教学楼 权重: 201
=====
最小生成树总权重: 1101

按任意键继续....
```

②克鲁斯卡尔(Kruskal)算法功能截图

```
#####
                校园管道系统最优布设
#####
                1 普里姆(Prim)算法
                2 克鲁斯卡尔(Kruskal)算法
                0  返回主菜单
#####

请选择(0~2): 2

克鲁斯卡尔(Kruskal)算法最小生成树:
=====
食堂 --- 宿舍  权重: 100
校门 --- 实验楼 权重: 120
图书馆 --- 食堂 权重: 150
体育馆 --- 操场 权重: 150
操场 --- 校门  权重: 180
宿舍 --- 校门  权重: 200
图书馆 --- 教学楼 权重: 201
=====
最小生成树总权重: 1101

按任意键继续....
```

## 6、课程设计总结

通过本次《数据结构》课程设计，我独立完成了“武汉东湖学院校园导游系统”的规划、设计与实现。该系统以图结构为核心，综合运用了图的邻接矩阵与邻接表两种存储结构，并实现了多种经典图算法，全面满足了校园导游与管道布设的实际需求。

### 一、主要工作内容

系统设计与实现：完成了系统七大功能模块的开发，包括景点地图管理、景点与交通数据维护、信息查询、游览线路规划、最短路径查询及管道最优布设。

数据结构应用：实现了图的邻接矩阵（MatrixGraph）和邻接表（AdjGraph）两种存储结构，并在此基础上完成了 DFS、BFS 遍历，Dijkstra、Floyd 最短路径算法，以及 Prim、Kruskal 最小生成树算法的编码与调试。

功能测试与验证：对各模块进行了全面测试，确保景点与道路的增删改查、文件存储与读取、各类路径规划与优化功能均运行正确，系统稳定可靠。

### 二、遇到的问题与解决方法

图的存储与转换：在实现邻接矩阵向邻接表转换时，需注意边的双向存储与权重传递。通过仔细设计 MatrixToAdjList 函数，并配合 ArcNode 链表的构建，最终实现了正确且高效的转换。

算法调试与路径输出：在实现 Dijkstra 和 Floyd 算法时，路径输出格式与中间顶点的回溯较为复杂。通过增加 Disppath 和 Dispath 函数，并配合 apath 数组进行路径回溯，最终实现了清晰的最短路径输出。

最小生成树算法的实现：在 Kruskal 算法中，需实现并查集以判断环的存在。通过设计简单的路径压缩与集合合并逻辑，配合冒泡排序对边按权值排序，最终正确生成了最小生成树。

### 三、收获与体会

**理论与实践结合：**本次设计让我将课堂上学到的图存储、遍历、最短路径、最小生成树等抽象算法，应用于具体的校园导览与管道布设场景中，深刻体会到数据结构在解决实际问题中的强大作用。

**系统设计与模块化思维：**通过自顶向下的系统分析与模块划分，我进一步掌握了大型程序的架构方法，增强了代码的可读性、可维护性与可扩展性。

**调试与优化能力：**在实现各类算法过程中，我学会了如何使用测试数据逐步验证、使用输出调试定位问题，并对时间与空间复杂度有了更直观的认识。

**文档撰写与表达：**通过撰写详细的设计报告、绘制模块结构图、整理测试截图，我提升了技术文档的整理与表达能力，这对未来的项目开发与学术交流具有重要意义。

### 提示：

1、请同学们在本课程设计报告中填写好表头的专业、姓名及学号等信息后，需要完成本文档红色标注内容所示的任务；

2、将本课程设计报告电子稿命名为“学号\_姓名\_KS5.doc”，以附件形式发送到老师的指定邮箱（[765262360@qq.com](mailto:765262360@qq.com)）；

3、将本课程设计报告打印一份并装订好，在课程结束时（2025.12.12，第十七周周五）由学习委员一并收齐后交给老师；